



Additional Admin Topics

Help Documentation

Additional Admin Topics

Automating Login to SmarterStats

This HTML code below demonstrates how you can make a text link (e.g. "See Your Stats") automatically log a person in to the SmarterStats application. By putting a hidden form on a simple web page, you can fill in the "Site ID", "Username,, and "Password" information either via hard coding the data or through a scripting language like ASP, ASP.Net or ColdFusion.

For the example code listed below, we have the form values set to generic text (e.g. "put_their_siteid_or_domain") to show where you would hard code values that are submitted to the login.aspx page. You could also dynamically generate these values using a scripting language like ASP or ColdFusion (a sample ASP script would substitute value="put_their_siteid_or_domain" with value=<% =txtSiteID %>). The form action shown (<http://127.0.0.1:9999/smarterstats/login.aspx>) uses the default location of the SmarterStats Web Interface. If you have created a separate web site for SmarterStats, or assigned a different IP address for SmarterStats within IIS, this action would have to be altered to reflect this change. This example demonstrates how easy and powerful the SmarterStats application is in allowing companies to automate entry into the application.

```
<html>
```

```
<head> <meta http-equiv= "Content-Language" content= "en-us" > <meta http-equiv= "Content-  
Type" content= "text/html; charset=windows-1252 "> <title>SmarterStats Login</title> </head>
```

```
<SCRIPT LANGUAGE= "JavaScript" > function GoToStats() { document.statsform.submit(); }  
</SCRIPT>
```

```
<body>
```

```
<form name= "statsform" action= "http://127.0.0.1:9999/Login.aspx" method= "post" > <input type=  
"hidden" name= "shortcutLink" value= "autologin" id= "shortcutLink" > <input type= "hidden"  
name= "txtSiteID" id= "txtSiteID" value= "put_their_siteid_or_domain" > <input type= "hidden"  
name= "txtUser" id= "txtUser" value= "put_their_username" > <input type= "hidden" name= "txtPass"  
id= "txtPass" value= "put_their_password" > </form>
```

```
<p><a href= "JavaScript:GoToStats()" > Log into your stats </a></p>
```

```
</body>
```

```
</html>
```

SmarterStats Web Services

SmarterStats was built with custom configuration in mind. In addition to being able to customize the look and feel of SmarterStats, developers and/or System Administrators have the ability to code to the SmarterStats application using several different web services. These web services allow developers and/or System Administrators to automate a variety of different things: add sites to the Web Interface on the fly, grab site-specific bandwidth usage for billing purposes, set details on a specific site or server, update site information, test servers added to the Web Interface, and more.

To view the web services, and their corresponding functions, available to you, go to your default Web Interface install location and append "services/services_page.aspx". Your particular SmarterStats location can be the specific IP address of the Web Interface (e.g. if you installed SmarterStats with the default installation you would use <http://127.0.0.1:9999/services/SiteAdmin.aspx>, etc. or a specific IP address you assigned for the Web Interface if you custom-installed SmarterStats or moved SmarterStats to a new web server) or a specific site name if you configured SmarterStats as an actual website within IIS (e.g. smarterstats.yourdomain.com). Below you will find a brief explanation of the web services available to you, along with the default installation paths to the specific web services details page.

Server Admin Web Service

<http://127.0.0.1:9999/services/serveradmin.aspx>

There are six (6) functions of the service available for customization. These services allow developers and/or System Administrators to pull information relating to a server or servers from a custom page using SOAP, an HTTP GET command, or an HTTP POST command (examples for each are provided as well). The functions, and their corresponding properties, are:

- **GetServers** - returns a list of all servers running SmarterStats Enterprise Edition .
- **GetGlobalDefaults** - returns the global default settings for a specific server. These default settings include server IP address, server status, default file, document and download extensions, and more.
- **PingServer** - allows you to ping a specific server and see the results.
- **DeleteServer** - allows you to delete a server from the Web Interface.

- AddServer - allows you to add a server directly in to the Web Interface.
- UpdateServer - allows you to change server configurations on the fly.

Site Admin Web Service

<http://127.0.0.1:9999/services/siteadmin.asmx>

There are nine (9) functions of the service available for customization. These services allow developers and/or System Administrators to pull information relating to a particular site or sites from a custom page using SOAP, an HTTP GET protocol, or an HTTP POST command (examples for each are provided as well). Constants for these web services can be found here . These functions, and their corresponding properties, are:

- AddSite - allows you to add sites to the Web Interface on the fly.
- DeleteSite - allows you to delete a site from the Web Interface on the fly.
- GetAllSites - returns a listing of all sites being monitored by the Web Interface.
- GetSitesForServer - returns sites from a particular Server ID.
- moveSite - allows you to initiate a move command to the service for a particular site. This would allow you to automate moves within a control panel application.
- getSiteStatus - allows you to check the site status for a particular Site ID.
- UpdateSite - allows you to update information pertaining to a particular site that resides in the Web Interface.
- AddSiteWithFTP - Allows you to add a site with FTP logs directly in to the Web Interface.
- GetSite - returns a single site that you specify and that is listed in the Web Interface.

Bandwidth Web Service

<http://127.0.0.1:9999/services/bandwidth.asmx>

There are two (2) functions of the service available for customization. These services allow developers and/or System Administrators to pull information relating to bandwidth usage for a particular site or server from a custom page using SOAP, an HTTP GET command, or an HTTP POST command (examples for each are provided as well). These functions can be extremely useful when providing bandwidth statistics to end users or for bandwidth monitoring and overage billing. The functions, and their corresponding properties, are:

- GetForSite - returns all bandwidth for a particular site.
- GetForServer - returns all bandwidth for a particular server on a per-site basis.

Product Information Web Service

<http://127.0.0.1:9999/services/productinfo.asmx>

There are four (4) functions of the service available for customization. These services allow developers and/or System Administrators to pull information (e.g. licensing info) or set information (e.g. license activation) relating to the product installed on the server from a custom page using SOAP, an HTTP GET command, or an HTTP POST command (examples for each are provided as well). The functions, and their corresponding properties, are:

- `ActivateLicenseKey` - allows you to activate the license key purchased for your product.
- `GetLicenseInfo` - returns information pertaining to the current product license.
- `GetProductInfo` - returns information pertaining to the currently installed product.
- `SetLicenseKey` - sets a license key for the Web Interface.

User Administration Web Service

<http://127.0.0.1:9999/services/useradmin.asmx>

There are seven (7) functions of the service available for customization. These services allow developers and/or System Administrators to perform various user functions, such as user creation and deletion, updating user information, retrieving user information and more from a custom page using SOAP, an HTTP GET command, or an HTTP POST command (examples for each are provided as well). The functions, and their corresponding properties, are:

- `AddUser` - automatically add users to a specific site.
- `SetSiteOwner` - set site ownership for a specific site to a specific user.
- `GetUser` - returns a user for a particular site.
- `ValidateLogin` - returns login results for a particular user.
- `GetUsers` - returns all users for a specific site.
- `UpdateUser` - updates user information for a site.
- `DeleteUser` - deletes the specified user from a site.

Enhancing Geographic Reports

SmarterStats uses a very efficient and accurate Geographic engine based on databases provided by MaxMind. These databases are very accurate, and correctly identify most traffic hitting your server. Because geo-location data changes frequently, SmarterStats will release a new update of these databases with every minor release of SmarterStats.

For those requiring constantly up-to-date geo-location data, MaxMind makes available a premium database that is updated monthly. In addition, the premium database takes into account dial-up services like AOL and Earthlink.

[Click here for the MaxMind premium database compatible with SmarterStats](#)



For questions regarding the premium MaxMind GeoIP Region Database, please click here .
SmarterTools does not directly support the MaxMind databases.

Instructions for installing premium database updates

- Download the latest Binary Format distribution of the Region database
- Using a product like WinZip or WinRar, extract the file to a temporary directory
- Locate a file named something similar to GeoIPRegion-###.dat and rename it to GeoIPRegion.dat
- Copy the file to your C:\program files\SmarterTools\SmarterStats\Service directory

SmarterStats allows you to customize its look and feel to suit your own needs. Instructions on skinning the web interface can be found below, as well as some notes that are important to understand.

Skinning Process

- Download the base skin module for SmarterStats. Download here
- Extract the contents into a temporary folder
- Modify the files using any of the common HTML / CSS editors on the market to adapt it to the look you want. Keep all tags that are surrounded by [brackets]. These are where controls will be added that contain the sidebar elements, body content, etc. For more details on each of the files and tags, see the section below.
- If desired, change any of the images in the Images directory
- Edit the Skin.xml file with notepad or an XML editor and modify the following:
 - Title - Input the title of your skin
 - Author - Input your name, or the name of your company
 - Save all files and make a new zip archive that contains all of the files. The structure of the zip file needs to remain the same as the skinbase.zip file you downloaded.
 - Login to SmarterStats as the administrator
 - Upload your skin in Settings - Skins, which will make it available to select as an active skin.

Restrictions

- The name of the product (SmarterStats) must remain visible in the skin. You may use a term like "Powered by", but the reference to SmarterStats must exist on the page.
- The [Copyright] tag must exist in the page, and must remain visible in the skin.

Skin Files

- Skin.xml - This XML file establishes the general properties of the skin, most notably the author and name of the skin. In the event that you give your skin to others, this helps to identify who made the skin.
- Login.htm - The template used for the display of the login page
- Plain.htm - The template for the inner frame where messages are viewed, as seen when reading a message
- Standard.htm - The template that is seen most of the time in SmarterStats, with a sidebar and a top menu
- SysAdmin.htm - The template that is seen most of the time in SmarterStats when logged in as the system administrator, with a sidebar and a top menu
- Calendar.css - This style sheet establishes the look and feel of the popup calendar and side calendar
- Layout.css (and Layout_print.css) - General layout styles are contained in this style sheet, including colors, etc. You may modify the styles, but do not delete any styles from this file to ensure proper operation of SmarterStats.
- LayoutPlain.css - Layout styles for the Plain.htm file are contained in this file. You may modify the styles, but do not delete any styles from this file to ensure proper operation of SmarterStats.
- Skin.css (and Skin_print.css) - Skin-specific styles reside in this file. You may alter it as you see fit

Bracket Tags

- [PAGETITLE] - Inserts the title of the page. This is most frequently used between <title> tags.
- [HEADITEMS] - Required placeholder for additional items that SmarterStats needs to add to each page. Must be contained withing <head> tags.
- [SIDEBAR] - Sidebar boxes
- [PRODUCTNAMEANDVERSION] - Text for "SmarterStats Professional Edition" + version. It is required to use this tag in the skin.
- [DOMAINTITLE] - The name of the domain that is logged in to.
- [TIPTTEXT] - Placeholder for where errors and messages will appear in response to actions
- [BODY] - Placeholder for the main inner content of the page. This is the big white area in the default skin.
- [COPYRIGHT] - This field is also required to be included and visible in the skin. Displays the SmarterStats copyright.

- [FOOTERCONTENT] - Used by SmarterStats internally. This tag must remain in all files in which it appears
- [MENU] - Placeholder for the main pull-down menu